ITI8700: Knowledge Representation

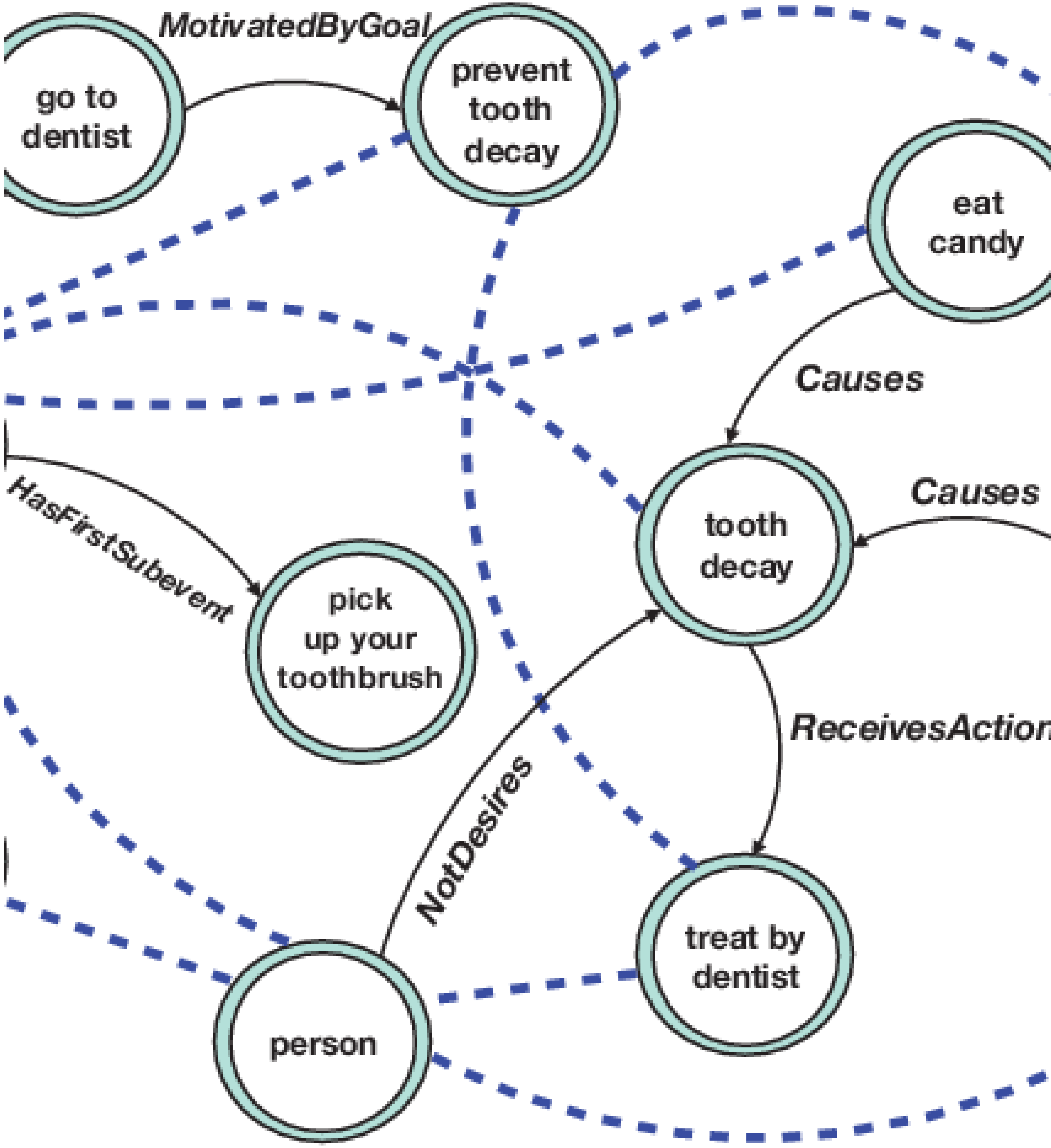
# Introduction to Semantic Parsing

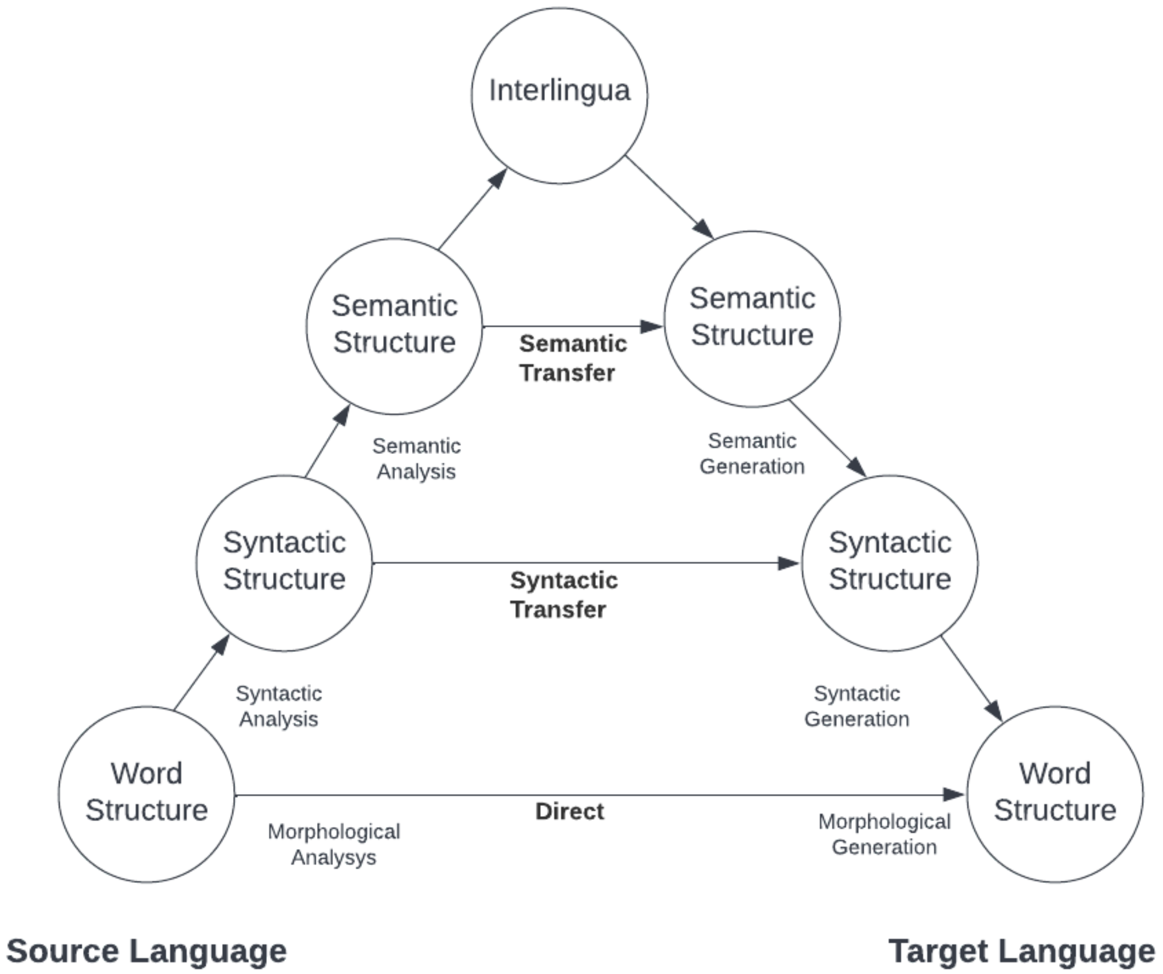
Martin Verrev

Spring 2024

# Commonsense Knowledge

Facts about the everyday world that a typical seven-year-old is expected to know including but not limited to space, time, objects, substances, environment, human psychology, societal norms, etc.

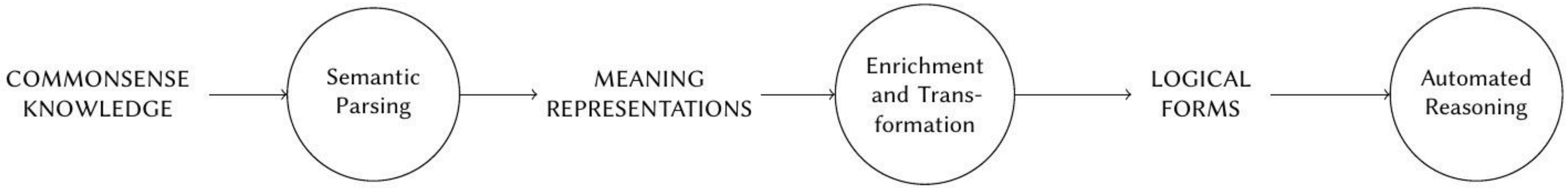





Bernard Vauquois. 1968

# Semantic Parsing

The task of converting natural language utterances to formal, machine-understandable representations of their meaning.



# A Semantic Parser for English

Why hard:

- Ambiguity: "I saw the man on the hill with the telescope"
- A large number of possible syntactic parse trees
- Roles of objects in the sentence hard to understand: subject, object, using, helper, place, time, …. (“A man opened the door” vs “A key opened the door”)
- Coreference resolution (“He saw it”)
- How to represent context/space/time etc?

# Additional Challenges

- Fractured landscape
- No universal representation schemes
- No unified validation metrics - Maximum Common Edge Subgraph Isomorphism; Elementary Dependency Match; SMATCH; Precision, Recall, and F1 etc for different frameworks.
- Not enough annotated data

# Semantic Parsing by Depth

- **Shallow** (Semantic Role Labelling) is concerned with identifying entities in an utterance and labelling them with the roles they play.
- **Deep** (Compositional Semantic Parsing) is concerned with producing precise meaning representations of utterances that can contain significant compositionality

# Semantic Parsing by Anchoring Relation

- For **Bi-lexical** dependency graphs, the graph nodes correspond to surface lexical units.
- **Anchored** semantic graphs are characterized by relaxing the correspondence relations between nodes and tokens while still explicitly annotating the correspondence between nodes and parts of the sentence.
- For **unanchored** dependency graphs, the correspondence between the nodes and tokens is not explicitly annotated

# Semantic Roles (Thematic Relations)

A semantic role is the underlying relationship that a participant has with the main verb in a clause.

| Semantic Role | Definition |
|---|---|
| AGENT | The volitional causer of an event |
| EXPERIENCER | The experiencer of an event |
| FORCE | The non-volitional causer of the event |
| THEME | The participate most directly affected by an event |
| RESULT | The end product of an event |
| CONTENT | The proposition or content of a propositional evnet |
| INSTRUMENT | An instrument used in an event |
| SOURCE | The origin of the object of a transfer event |
| GOAL | The desination of an object of a transfer event |
| BENEFICIARY | The beneficiary of an event |

# Data Sources for Semantic Role Labels

**Propbank** (proposition bank) is a corpus that is annotated with verbal propositions and their arguments plus a lexicon defining those argument roles on a per-verb basis. https://propbank.github.io/

**Unified Verb Index** is a system which merges links and web pages from four different natural language processing projects: VerbNet, PropBank, FrameNet, Ontonotes (https://verbs.colorado.edu/verb-index/)

**AMR Annotation Dictionary** is a resource that describes AMR specific annotations, e.g. `regardless-91`: https://www.isi.edu/~ulf/amr/lib/amr-dict.html

## Desired Attributes for Sematic Notation

1. **Verifiability.** System's ability to compare the state of affairs as described by a representation to the state of affairs as modeled in a knowledge base.
2. **Unambigious representation.** Words and sentences have different meaning representations in different contexts, but representations cannot be ambiguous.
3. **Canonical Form.** Distinct inputs meaning the same thing but having differents lexical utterances must have the same representation.
4. **Inference.** System must draw valid conclusions based on meaning of inputs and prior knowledge - propositions not explicitly stated but still logically derivable.
5. **Expressivness.** enough to handle a wide range of subject matter - ideally any sensible natural language utterance.

# Many Different Annotations Schemes ..

- Abstract Meaning Representation (AMR)
- Universal Conceptual Cognitive Annotation (UCCA)
- Universal Dependencies (UD)
- Elementary Dependency Structures (EDS)
- Prague Tectogrammatical Graphs (PDT)
- Discourse Representation Structures (DRS)
- Universal Decompositional Semantics (UDS)

.. and many good parsers.

- But none that is ideally suited for our task.

# Overview of Representation Schemes

| Name | Unit of Annotation | Flavor | Format | Primary Languages |
|---|---|---|---|---|
| **AMR** | sentence | unanchored | Penman | English (+Chinese) |
| **UCCA** | sentence | anchored | XML | English (+4 others) |
| UD | sentence | bi-lexical | CoNLL-U | multilingual |
| EDS | sentence | anchored | DAG | English |
| PTG | sentence | anchored | DAG | English, Czech |
| **DRS** | passage | anchored | nested boxes | English |
| **UDS** | sentence | anchored | Predicates + UD (CoNLL-U) | English |

# Overview of Parsers

| Notation | Parser | Platform | Stars | Forks | Commits | Last Commit |
|---|---|---|---|---|---|---|
| AMR | JAMR[6] | scala | 192 | 50 | 825 | March 2019 |
| AMR | Transition AMR parser[7] | python | 144 | 35 | 1838 | November 2022 |
| **AMR** | **amrlib**[8] | python | 146 | 22 | 166 | March 2022 |
| UCCA | UCCA parser[9] | python | 18 | 7 | 6 | June 2019 |
| **UCCA** | **TUPA**[10] | python | 73 | 22 | 2135 | December 2020 |
| UD | UDepLambda[11] | java | 85 | 22 | 225 | July 2018 |
| UD | uuparser[12] | python | 77 | 26 | 125 | October 2020 |
| UD | stanza[13] | python | 6400 | 830 | 3146 | September 2022 |
| EDS | Pydelphin[14] | python | 68 | 24 | 1043 | October 2022 |
| EDS | HRG Parser[15] | python,java | 9 | 0 | 4 | October 2018 |
| PTG | Perin[16] | python | 41 | 5 | 36 | Oct 04 2021 |
| **DRS** | **TreeDRSparsing**[17] | python | 5 | 2 | 59 | March 2020 |
| DRS | EncDecDRSParsing[18] | python | 36 | 11 | 15 | August 2019 |
| **UDS** | **Predpatt**[19] | python | 110 | 23 | 59 | February 2021 |
| UDS | MISO[20] | python | 7 | 1 | 1019 | September 2021 |

# Our approach to semantic parsing: DIY

- We generate individual AMR parse trees from passage.
- We take the output of the Stanza Universal Dependencies Parser (Stanford)
- We apply UD parse as a constraint for AMR parse trees to:
    - classify the sentence
    - perform "sanity check" for AMR graph to identify and attempt to fix inconsistencies.
- We construct the sentence (and passage) level context objects.
- We apply sentence based semantic role mappings: ARG0 -> agent (from Propbank), mapping for custom AMR predicates: ‘have-org-role-91‘ -> ‘role‘
- And then convert the result to 1st-order logic.

# Universal Dependencies

Cross-linguistically consistent treebank annotation for many languages, with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective.

The annotation scheme is based on an Stanford dependencies, Google universal POS tags and the Interset interlingua.

See also: https://universaldependencies.org

# UD Example

Cats are animals.

```
root: animal [id:3 text:animals upos:NOUN xpos:NNS feats:Number=Plur]
  nsubj: cat [id:1 text:Cats upos:NOUN xpos:NNS feats:Number=Plur]
  cop: be [id:2 text:are upos:AUX xpos:VBP feats:Mood=Ind|Tense=Pres|VerbForm=Fin]
  punct: . [id:4 text:. upos:PUNCT xpos:.]
```

# Universal Conceptual Cognitive Annotation (UCCA)

The central idea of the project is to analyze and annotate natural languages using purely semantic categories and structure (a graph). Syntactic categories and structure are not part of the manual annotation, and are ideally learned implicitly by the parsers.

Base element of the foundational layer is a *scene* - describing movement, action or event with optional spatial and temporal relations.

https://universalconceptualcognitiveannotation.github.io/

**Demo (currently offline):** http://ucca-demo.cs.huji.ac.il/

# UCCA Foundational Layer

| Abb. | Category | Short Definition |
|---|---|---|
| **Scene Elements** | | |
| P | **Process** | The main relation of a Scene that evolves in time (usually an action or movement). |
| S | **State** | The main relation of a Scene that does not evolve in time. |
| A | **Participant** | A participant in a Scene in a broad sense (including locations, abstract entities and Scenes serving as arguments). |
| D | **Adverbial** | A secondary relation in a Scene (including temporal relations). |
| **Elements of Non-Scene Units** | | |
| C | **Center** | Necessary for the conceptualization of the parent unit. |
| E | **Elaborator** | A non-Scene relation which applies to a single Center. |
| N | **Connector** | A non-Scene relation which applies to two or more Centers, highlighting a common feature. |
| R | **Relator** | All other types of non-Scene relations. Two varieties: (1) Rs that relate a C to some super-ordinate relation, and (2) Rs that relate two Cs pertaining to different aspects of the parent unit. |
| **Inter-Scene Relations** | | |
| H | **Parallel Scene** | A Scene linked to other Scenes by regular linkage (e.g., temporal, logical, purposive). |
| L | **Linker** | A relation between two or more Hs (e.g., “when”, “if”, “in order to”). |
| G | **Ground** | A relation between the speech event and the uttered Scene (e.g., “surprisingly”, “in my opinion”). |
| **Other** | | |
| F | **Function** | Does not introduce a relation or participant. Required by the structural pattern it appears in. |

# UCCA Example

Cats are animals.

```
corpus-min_0000:
  Light verbs:
    1.22->1.24 [F are]
  Predicate nouns:
    1.21->1.22 [P [C Cats] [F are] ]
  Center:
    1.1->1.2 [C animals]
    1.22->1.23 [C Cats]
  Punctuation:
    1.1->1.3 [U .]
  Participant:
    1.1->1.22 [A [C Cats] [F are] ]
    1.1->1.23 [A Cats]
  Function:
    1.22->1.24 [F are]
```

# Discourse Representation Structures (DRS)

DRS is a representation scheme based on Discourse Representation Theory. In contrast to ordinary treebanks, the units of annotation in the corpus are texts rather than isolated sentences.

Basic DRSs consist of discourse referents like $x$ representing entities and discourse conditions like $man(x)$ representing information about discourse referents. The corpus is based on Groningen Meaning Bank that annotates English texts with formal meaning representations rooted in Combinatory Categorial Grammar.

See also: LangPro. https://naturallogic.pro/LangPro/

# DRS Example

Cats are animals.

```
DRS-0(
    cat( X1 )
    animal( X1 )
    be( E1 )
    Agent( E1 X2 )
    Patient( E1 X3 )
    now( T1 )
    Temp_included( E1 T2 )
    Equ( T2 T2 )
)
```

# DRS Example

(28)

| $x_1, y_1$ |
| --- |
| policeman($x_1$)<br>squirrel($y_1$)<br>chase($x_1, y_1$) |

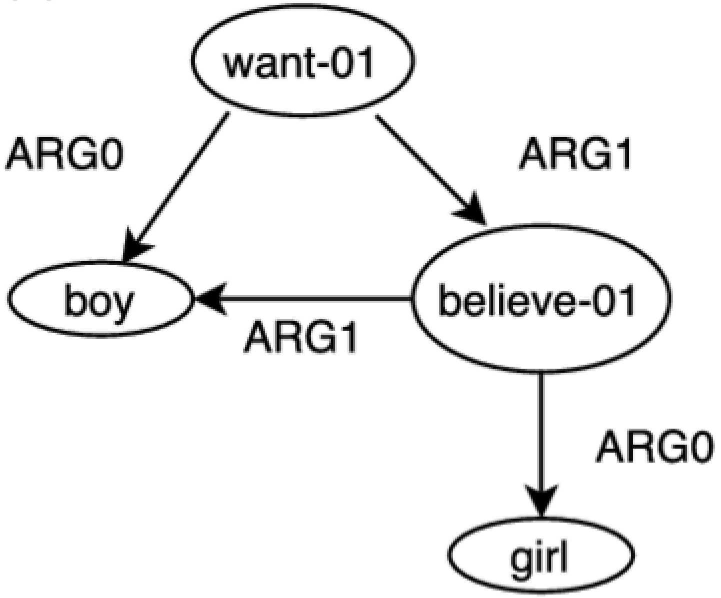
DRS can be viewed as a first-order language without explicit quantifiers.

# AMR

Abstract Meaning Representation is a semantic formalism based on propositional logic and the neo-Davidsonian event representations where each representation is a single-rooted, directed graph. AMR is strongly biased towards English though it does support multilingual meanings. Its concepts are either English verbs, PropBank framesets, or specific keywords. AMR also supports NER, question detection, within-sentence co-reference, modality and question identification.

# AMR Representations

(a)




Graph notation

(b)

```
(w / want-01
   :ARG0 (b / boy)
   :ARG1 (b2 / believe-01
            :ARG0 (g / girl)
            :ARG1 b))
```

Penman notation

# Benefits of AMR

- Tools for processing: https://penman.readthedocs.io/en/latest/api/penman.html
- Can be represented as triples: manageable post-processing
- Penman notation both human and machine-readable.
- Question detection with `amr-unknown` keyword.
- Additional information extracted as compared to UD:
    - Role reification
    - NER and nunber detection
    - Tone detection
    - Wikification

    .. etc

# Penman notation

```
┌──────────────────────────── Variable (this one is the graph's top)
│     ┌────────────────────── Instance relation
┴ ────┴────
(d / drive-01
  ┬ ───┬────
  │    └───────────────────── Concept (node label)
  └────────────────────────── Indicates the node's concept
        ┌──────────────────── Edge relation
──────┴──────
   :ARG0 (h / he)
──┬──
  └────────────────────────── Role (edge label)
   :manner (c / care-04
                  ┌────────── Attribute relation
           ─────┴─────
           :polarity -))
                     ┬
                     └─── Atom (or "constant")
```

# Example

Brutus stabs Caesar with a knife.

## AMR parse graph

```
(s / stab-01
      :ARG0 (p / person
            :name (n / name
                  :op1 "Brutus"))
      :ARG1 (p2 / person
            :name (n2 / name
                  :op1 "Caesar"))
      :instrument (k / knife))
```

# Thank you.